

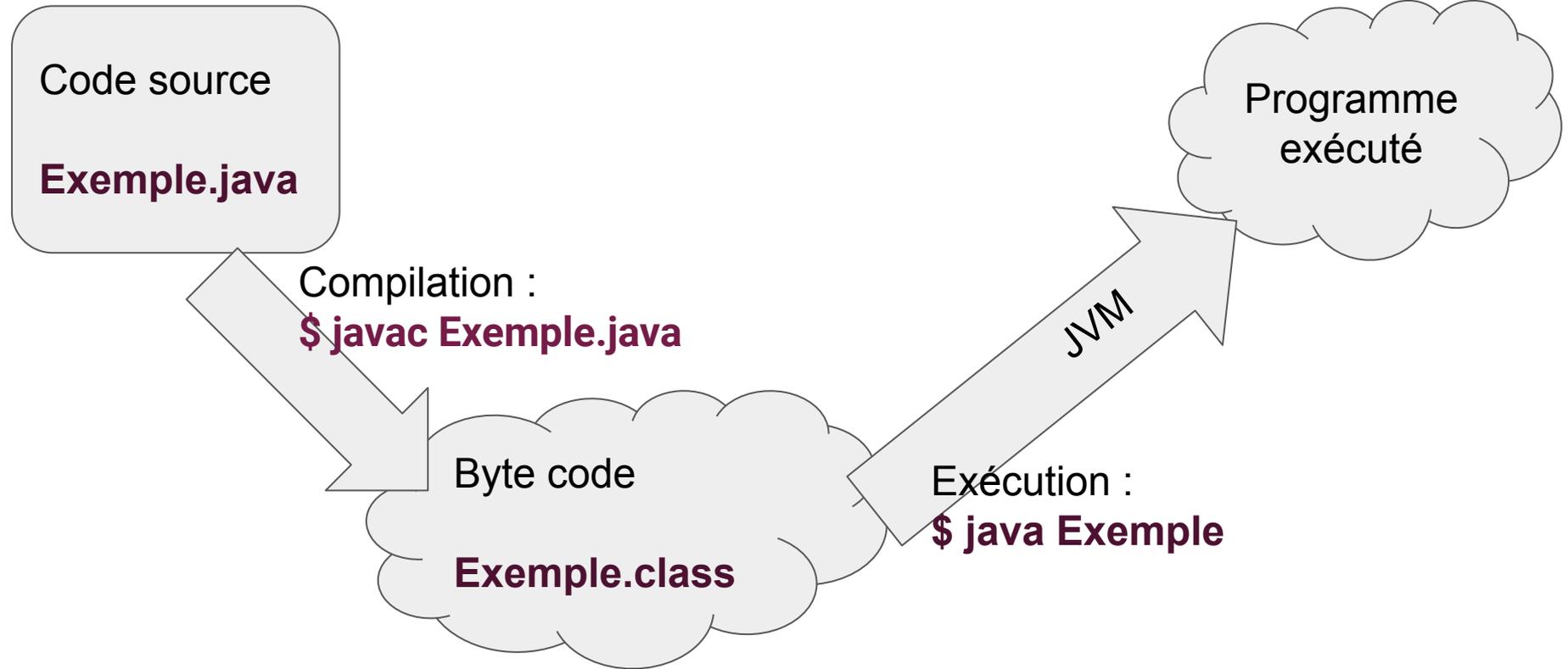
Environnement de développement

Info 211 IPO

Alice Jacquot

<https://www.lri.fr/~jacquot/ipo>

Java : un langage semi-compilé



RAM : que se passe-t-il à l'exécution ?

Heap (touche pas) (en C++ si, exécutable direct, gestion mémoire à faire direct dans code)

Heap Managé (là que ça se passe pour nous) Notre appli tourne sur une partie : utilise des blocks, à libérer
: Gestion de la mémoire



Stack : s'autogère, contient les références

JIT (JIT compiler) (*Just-In-Time*): Transforme .class (contenu sur le disque) en langage machine **et** le monte en mémoire (RAM) (surtout en heap managé, un peu en stack) **au moment nécessaire**

La JVM(Machine Virtuelle Java)

- Application qui tourne
- Alloue de la mémoire
- JVM décide quoi et quand allouer
- Garbage Collector tourne en tâche de fond pour savoir qui libérer
 - En java on crée des objets en mémoire via new (une référence qui pointe vers l'objet). Si un objet n'a pas de référence associés, il est donc tagué comme supprimable et cible du garbage collector
 - On peut avoir plusieurs référence sur un même objet : utilisation d'un compteur de référence. Supprimable = compteur à 0.
- Allocation + suppression = gestion de la mémoire.

À quoi ressemble le code source ?

- Contenu dans des fichier .java
- Une classe = un fichier
- Nom du fichier = nom de la classe (avec une majuscule Exemple.java contient le code de la classe Exemple)
- La classe principale contient la méthode static main :
public static void main(String[] args)

Le fichier **Exemple.java**

```
public class Exemple {  
  
    public static void main(String[] args) {  
        String myString = "Exemple";  
        afficheBonjour(myString);  
    }  
  
    private static void afficheBonjour(String s){  
        String autre = "Bonjour " + s.toUpperCase();  
        System.out.println(autre);  
    }  
  
}
```

JRE, JDK

Environnement de développement \neq Environnement d'exécution :

Pas la même chose installée chez le dev et chez l'utilisateur !

- JRE : Java Runtime Environment (machine virtuelle, bibliothèque logicielle, plugin navigateur)
- JDK : Java Development Kit (compilateur, javadoc, JRE “privé”, jar ...)

IDE

Integrated Development Environment (EDI en français)

Un IDE contient en général :

- un éditeur de texte adapté aux langages supportés
- des outils de compilation et d'exécution
- des outils de test
- des outils d'analyse de code
- de la documentation intégrée
- des outils de recherche et de navigation spécifique à la programmation (rechercher une méthode, une classe, la définition d'une variable...)

Quelques IDE

- Eclipse (Java, Multilangage)
- IntelliJ IDEA (Java, Scala)
- Visual Studio (C++, C#, .NET)
- Code::Blocks (C, C++, Fortran)
- ...

Avantage et inconvénients

- Tout en un : codage, compilation, exécution, gestion de version,...
- Analyse à la volée
- Coût d'apprentissage, d'installation, de configuration
- Configurations pérennes (options de compilation)
- Lourd : ce n'est pas un outil de déploiement !
- Peut planter (pertes de temps)
- Des outils et vues adaptés au langage et aux usages
- Hautement spécifique
- Refactorisation de code facilité
- ...

Eclipse : mastodonte historique

Existe depuis 2001, très répandu.

Libre

Ce que vous utiliserez “par défaut” en TP.

Multi-langage

Fonctionne avec des perspective en fonction du langage ou framework utilisé





AccessDB.java Person.java *Exemple.java

```
1 public class Exemple {
2
3
4 public static void main(String[] args) {
5
6     String myString = "Exemple";
7     affiche_bonjour(s);
8 }
9
10 private static void affiche_bonjour(String s){
11     System.out.println();
12 }
13
14 }
15
```

Task List

Find All Ac

Outline

- Exemple
 - main(Strin
 - affiche bo

Problems @ Javadoc Declaration Console

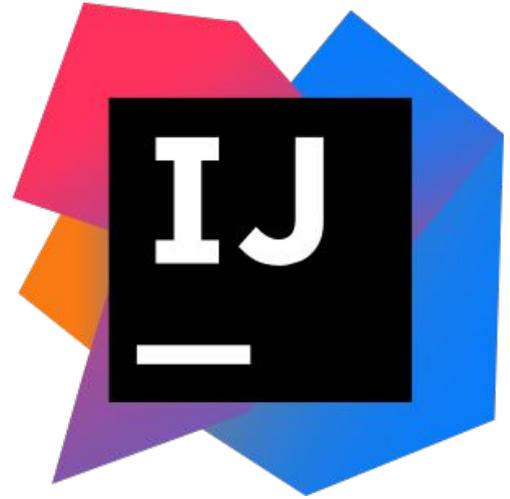
No consoles to display at this time.



IntelliJ

Existe depuis 2001, de plus en plus répandu

- Moins lourd
- version partielle gratuite et libre, version complète commerciale
- Similaire à eclipse pour les usages simples
- Spécifique Java
- Davantage orienté console et navigation clavier



Project

- core-api [intellij.platform.core]
 - src
 - com.intellij
 - codeinsight
 - concurrency
 - core
 - diagnostic
 - formatting
 - ide
 - injected.editor
 - lang
 - folding
 - CompositeFoldingBuilder
 - CustomFoldingBuilder
 - CustomFoldingProvider
 - FoldingBuilder
 - FoldingBuilderEx
 - FoldingDescriptor
 - LanguageFolding
 - injection
 - ASTNode
 - CodeDocumentationAwareCo
 - CodeDocumentationAwareCo
 - Commenter
 - CompositeLanguage
 - CustomUncommenter
 - DependentLanguage
 - FCTSBackedLighterAST
 - FileASTNode
 - InjectableLanguage
 - ITokenTypeRemapper
 - Language

```
49  /**
50   * Only queries base language results if there are no extensions for originally requested
51   */
52  @NotNull
53  @Override
54  public List<FoldingBuilder> allForLanguage(@NotNull Language language) {
55      for (Language l = language; l != null; l = l.getBaseLanguage()) {
56          List<FoldingBuilder> extensions = forKey(l);
57          if (!extensions.isEmpty()) {
58              return extensions;
59          }
60      }
61      return Collections.emptyList();
62  }
63
64  @NotNull
65  public static FoldingDescriptor[] buildFoldingDescriptors(@Nullable FoldingBuilder builder
66                  if (!DumbService.isDumbAware(builder) && DumbService.getInstance(root.getProject()).isDum
67                  return FoldingDescriptor.EMPTY;
68  }
69
70  if (builder instanceof FoldingBuilderEx) {
71      return ((FoldingBuilderEx)builder).buildFoldRegions(root, document, quick);
72  }
73  final ASTNode astNode = root.getNode();
74  if (astNode == null || builder == null) {
75      return FoldingDescriptor.EMPTY;
76  }
77
78  return
79  }
80 }
81
```

`builder.buildFoldRegions(ASTNode node, Document document) FoldingDescriptor[]`

`FoldingDescriptor.EMPTY (com.intellij.lang... FoldingDescriptor[]`

Dot, space and some other keys will also close this lookup and be inserted into editor >>

```

@GetMapping("/")
@ResponseBody
@Transactional(readOnly = true)
public List<City> helloWorld() {
    return cs.find;
}

```

  cityService.findCities(CitySearchCriteria criteria, Pageable pagea

^↓ and ^↑ will move caret down and up in the editor >>

Version Control

Version Control: Local Changes Log

Branch: All User: All Date: All Paths: All

use psiTraverser() in CheckPsiReadAccessors	Daniil Ovchinnikov	21.08.18, 15:23	intellij.pycharm.community.resources 1 file
remove unused code	Dmitry Batkovich	21.08.18, 15:03	resources/idea 1 file
IDEA-59397 When caret is outside visible editor area, on Alt-En	Dmitry Batrak	21.08.18, 15:01	PyCharmCoreApplicationInfo.xml
[gui-test] replace system properties with env variables	Vladislav Shishov	21.08.18, 14:12	
add a test case for java same parameter value inspection	Dmitry Batkovich	21.08.18, 14:16	
[groovy] parse new lines in type parameter list (IDEA-197524)	Daniil Ovchinnikov	21.08.18, 14:09	
cleanup for code review IDEA-CR-36249	Nikita Skvortsov	21.08.18, 14:01	
PyCharm 2018.1.5	 origin/181.5540	Aleksey Rostovskiy	21.08.18, 14:06
check for sa pid attach availability - IDEA-168185	Egor Ushakov	21.08.18, 14:01	PyCharm 2018.1.5
[groovy] wrap PsiFile#isValid check into a read action (IDEA)	 Daniil Ovchinnikov*	20.08.18, 21:05	8935cc3 Aleksey Rostovskiy < aleksey.rostovskiy@jetbrains.com > on 21.08.18 at 14:06
cleanup	 Alexey Kudravtsev	21.08.18, 12:28	 origin/181.5540  pycharm/181.5540.17

In 1 branch: origin/181.5540

Commencer à coder sous IntelliJ

Les codes et bytes codes produits sont regroupés par projet

- File/New/Project... puis next (pour un projet java standard), next, donner un nom au “projet” et finish

Les nouvelles classes se créent dans src

- Clic droit, new/Java Class

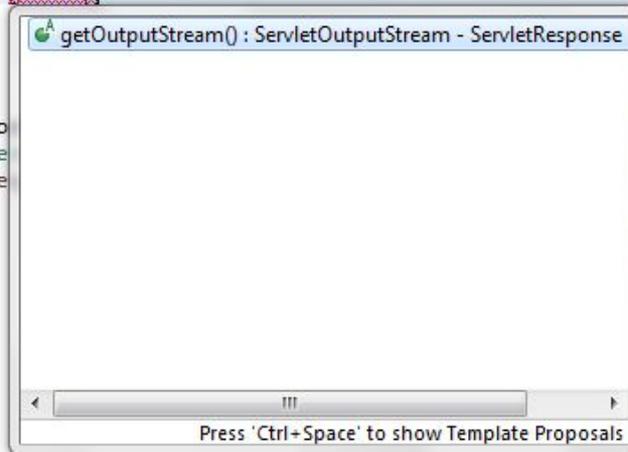
De même, les classes fournies doivent être copiées (les fichiers) dans src

Javadoc et completion

- Accès à la volée à la javadoc depuis n'importe où dans le code (clic ou survol de méthodes)
- L'autocomplétion rend facile l'utilisation de noms longs, suggère des paramètres

```
protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  
    // TODO Auto-generated method stub  
    var out = resp.getoutp  
}
```

```
@Override  
protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  
    // TODO Auto-generated method stub  
    super.doPost(req, resp);  
}
```



Returns a [ServletOutputStream](#) suitable for writing binary data in the response. The servlet container does not encode the binary data.

Calling `flush()` on the `ServletOutputStream` commits the response. Either this method or [getWriter](#) may be called to write the body, not both.

Returns:
a [ServletOutputStream](#) for writing binary data

Throws:
[IllegalStateException](#) - if the `getWriter` method has been called on this response
[IOException](#) - if an input or output exception occurred

See Also:

Press 'Tab' from proposal table or click for focus

Génération automatique de code

- Java est un langage verbeux
- Structure d'une classe, méthode main
- Plus tard : getter et setter, .toString, .equals, constructeurs...
Souvent à modifier, toujours à vérifier !
- Javadoc : à compléter !

```
public class Exemple {  
  
    int monAge;  
  
    public int getMonAge() {  
        return monAge;  
    }  
  
    public void setMonAge(int monAge) {  
        this.monAge = monAge;  
    }  
  
    public static void main(String[] args) {  
        String myString = "Exemple";  
        afficheBonjour(myString);  
    }  
  
    /**  
     * Affiche "Bonjour" suivi de la version  
     * tout en majuscules de s  
     *  
     * @param s  
     */  
    public static void afficheBonjour(String s){  
        String autre = "Bonjour " + s.toUpperCase();  
        System.out.println(autre);  
    }  
}
```

Pour aller (beaucoup) plus loin

Par environnement de développement, en entreprise, on parle aussi souvent de tout ce qui va autour du développement :

- Gestionnaire de projets (MAVEN, ...)
- Framework
- Outils de déploiement
- Uniformité du code (feuille de style)
- Outils d'analyse qualité
- Environnements de développement, de tests, de production
- ...